How to Utilize Side Information: Decomposed Gaussian Process Regression

Kai Wang, Bryan Wilder, Sze-Chuan Suen, Milind Tambe, Bistra Dilkina

Problem Background

- Goal: we want to minimize the total Tuberculosis infected population in the following 25 years.
- Input: policy $v \in R^n, n = 110$
  - Where v_i refers to what proportion of people in age i would catch the information about Tuberculosis (so that they will seek a treatment and get cured.)
- Budget constraint: $\sum v_i \leq B$

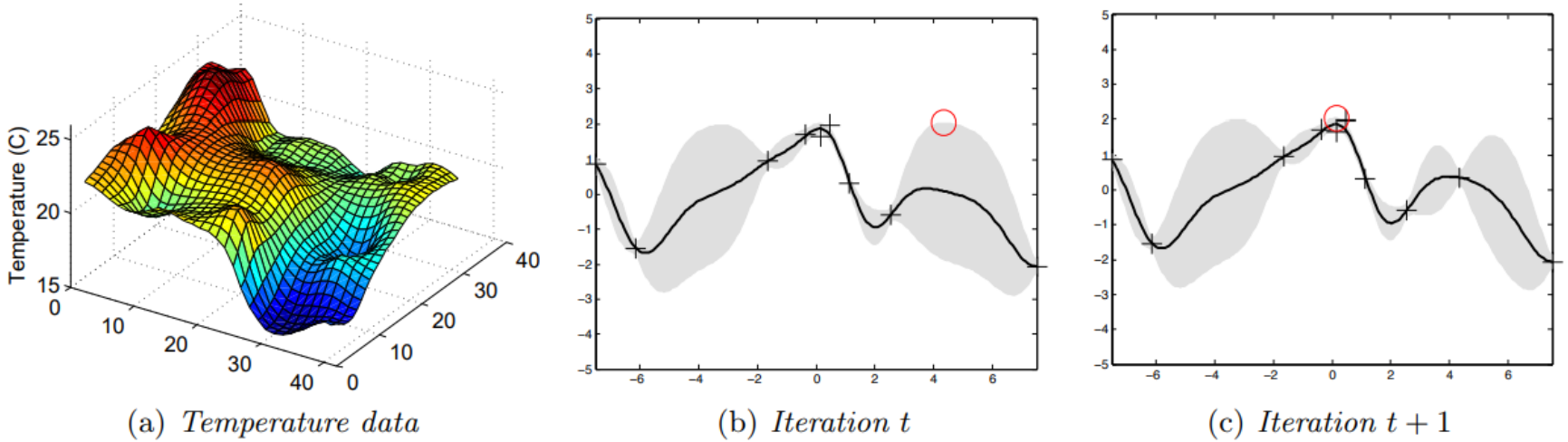
Problem Background

- What we have:
  - A very expensive simulation model (or a field test) that can simulate the interaction between people. Let's call this function f
  - $f(v)$ represents the outcome by applying policy v.
  - But it takes 30 minutes to run one $f(v)$ …

Gaussian Process Upper-confidence Based

$$\mu_T(x) = k_T(x)^T(K_T + \sigma^2 I)^{-1} y_T$$
$$k_T(x, x') = k(x, x') - k_T(x)^T(K_T + \sigma^2 I)^{-1} k_T(x')$$
$$\sigma_T^2(x) = k_T(x, x)$$

$$k_T(x) = [k(x_1, x), \ldots, k(x_m, x)]^T$$
$$K_T = [k(x, x')]_{x,x' \in A_T} \in S_{++}^n$$



(a) *Temperature data*

(b) *Iteration t*

(c) *Iteration t + 1*

Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design, 2010

In our problem

- We can use Gaussian process regression to approximate our $f(v), v \in R^n$
- But we don't know the kernel $k(x, x')$…

Markov property

- When we run a simulation, it will give us the total infected population of age i at time t, which is denoted by I_i^t.
- Also the healthy population S_i^t and latent TB population E_i^t
- Let $x_t = [I_i^t, S_i^t, E_i^t]$ be the collection of all variables at time t.

Markov property

- memoryless property
  - $E[x_{t+1}|x_t, x_{t-1}, \dots, x_1, v] = E[x_{t+1}|x_t, v]$
- So we can write down the variables at time $t+1$ as $h_t(x_t, v)$

Markov property

- memoryless property
  - $E[x_{t+1}|x_t, x_{t-1}, \dots, x_1, v] = E[x_{t+1}|x_t, v]$
- So we can write down the variables at time $t+1$ as $h_t(x_t, v)$
- Motivated by the SEIS population model:
  - $I_{i+1}^{t+1} = I_i^t(1-d_i^t)(1-\nu_i) + E_i^t\alpha_i^t$
  - $E_{i+1}^{t+1} = E_i^t(1-\mu_i^t)(1-\alpha_i^t) + S_i^t(1-\mu_i^t)\sum_{k=1}^n \beta_{ik}\frac{I_k^t}{I_k^t+E_k^t+S_k^t}$
  - $\approx E_i^t(1-\mu_i^t)(1-\alpha_i^t) + constant\,(1-\mu_i^t)\sum_{k=1}^n \beta_{ik}\frac{I_k^t}{constant}$

Linear approximation

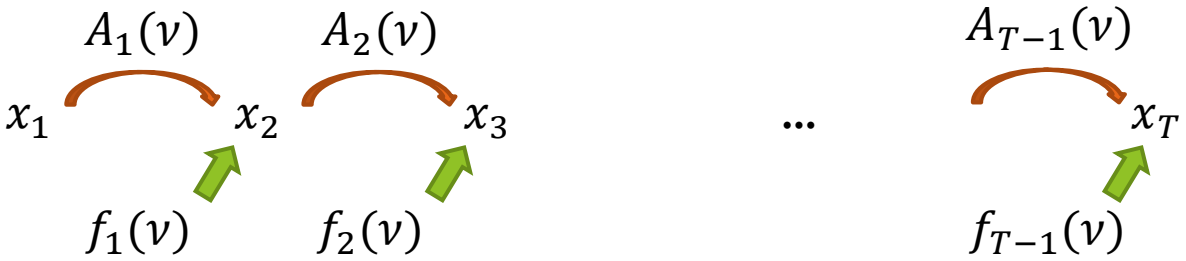
- $x_{t+1} = h_t(x_t, \nu) = A_t(\nu)x_t + f_t(\nu)$
  - Where $A_t(\nu) \in R^{n \times n}$ is a matrix of linear function in terms of ν
  - And $f_t(\nu)$ is the difference (error) of x_{t+1} and $A_t(\nu)x_t$

Linear approximation

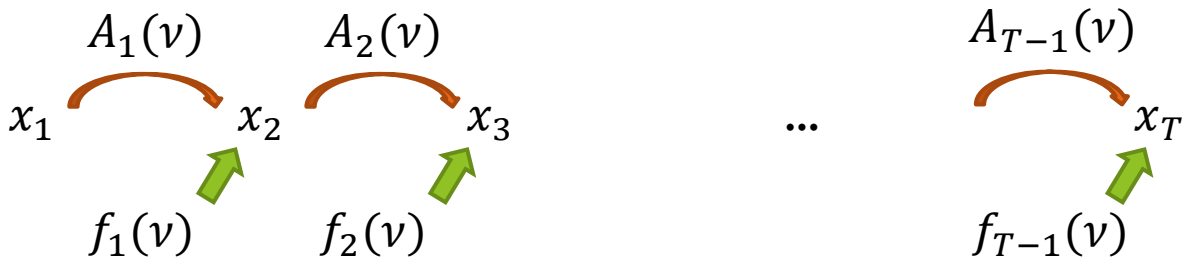
- $x_{t+1} = h_t(x_t, v) = A_t(v)x_t + \mathrm{f_t}(v)$
  - Where $A_t(v) \in R^{n \times n}$ is a matrix of linear function in terms of v
  - And $f_t(v)$ is the difference (error) of x_{t+1} and $A_t(v)x_t$
  - We assume this function to be a Gaussian process with certain kernel (e.g. radius based kernel)
  - So we can use Gaussian process regression to approximate this difference (error) function $f_t(v) = x_{t+1} - A_t(v)x_t$

Fragmentation model

Fragmentation model

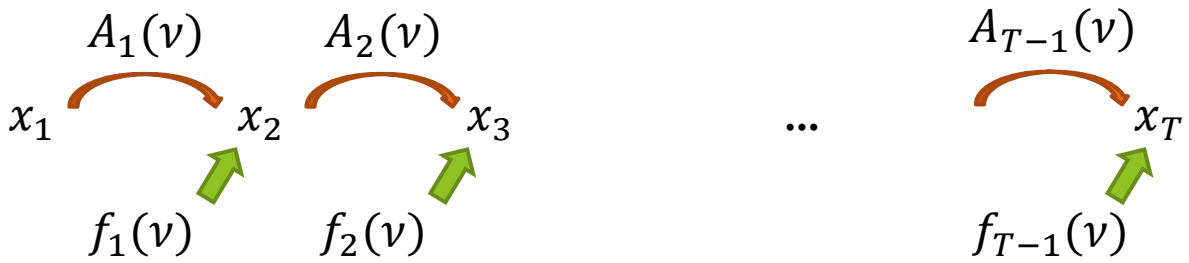


Fragmentation model



$$
\begin{aligned}
x_k &= A_{k-1}(\nu)x_{k-1} + f_{k-1}(\nu) \\
&= A_{k-1}(\nu)\big(A_{k-2}(\nu)x_{k-2} + f_{k-2}(\nu)\big) + f_{k-1}(\nu) \\
&= A_{k-1}(\nu)A_{k-2}(\nu)x_{k-2} + A_{k-1}(\nu)f_{k-2}(\nu) + f_{k-1}(\nu) \\
&\dots \\
&= p_1(\nu)f_1(\nu) + p_2(\nu)f_2(\nu) + \cdots + p_{T-1}(\nu)f_{T-1}(\nu) + p_0(\nu)x_1
\end{aligned}
$$

Fragmentation model



$$
\begin{aligned}
x_k &= A_{k-1}(\nu)x_{k-1} + f_{k-1}(\nu) \\
&= A_{k-1}(\nu)\big(A_{k-2}(\nu)x_{k-2} + f_{k-2}(\nu)\big) + f_{k-1}(\nu) \\
&= A_{k-1}(\nu)A_{k-2}(\nu)x_{k-2} + A_{k-1}(\nu)f_{k-2}(\nu) + f_{k-1}(\nu) \\
&\dots \\
&= p_1(\nu)f_1(\nu) + p_2(\nu)f_2(\nu) + \cdots + p_{T-1}(\nu)f_{T-1}(\nu) + p_0(\nu)x_1
\end{aligned}
$$

$$
\sum x_k = g_1(\nu)f_1(\nu) + g_2(\nu)f_2(\nu) + \cdots + g_{T-1}(\nu)f_{T-1}(\nu) + g_T(\nu)x_1
$$

Kernel

- So we know the total infected population in the following T years would be
  - $f(v) = 1_I^T \sum x_k = g_1(v)1_I^T f_1(v) + g_2(v)1_I^T f_2(v) + \cdots + g_{T-1}(v)1_I^T f_{T-1}(v) + g_T(v)1_I^T x_1$

Kernel

- So we know the total infected population in the following T years would be
  - $f(\nu) = 1_I^T \sum x_k = g_1(\nu)1_I^T f_1(\nu) + g_2(\nu)1_I^T f_2(\nu) + \cdots + g_{T-1}(\nu)1_I^T f_{T-1}(\nu) + g_T(\nu)1_I^T x_1$
- Since we are assuming each simpler Gaussian process to have kernel $k_i(\nu, \nu')$
- The entire kernel of function f is
  - $k(\nu, \nu') = g_1(\nu)k_1(\nu, \nu')g_1(\nu') + \cdots + g_T(\nu)k_T(\nu, \nu')g_T(\nu')$

Gaussian process regression

- Now we can finally do the Gaussian process regression…
- $f(v) = g_1(v)f_1(v) + g_2(v)f_2(v) + \cdots + g_T(v)f_T(v)$
- Then?

Gaussian process regression

- Now we have two options.
  - 1. run Gaussian process regression on the entire function $f(\nu)$ with cumulative kernel $k(\nu, \nu')$
  - 2. run Gaussian process regression on each $f_i(\nu)$ with kernel $k_i(\nu, \nu')$ then sum them up.

Short story

- Long story short, the second one is better.
  - 1. run Gaussian process regression on the entire function $f(\nu)$ with cumulative kernel $k(\nu, \nu')$
  - 2. run Gaussian process regression on each $f_i(\nu)$ with kernel $k_i(\nu, \nu')$ then sum them up.

Long story…

- Long story
  - We want to compare the variance derived from two different methods.
  - $var_{entire}(x) = k(x, x') - k_T(x)^T K_T^{-1} k_T(x')$
  - $var_i(x) = k_i(x, x') - k_{i,T}(x)^T K_{i,T}^{-1} k_{i,T}(x')$

$$k_T(x, x') = k(x, x') - k_T(x)^T (K_T + \sigma^2 I)^{-1} k_T(x')$$
$$\sigma_T^2(x) = k_T(x, x)$$

Long long story…

- $k_T(x, x') = k(x, x') - k_T(x)^T K_T^{-1} k_T(x')$
- $= \sum_i g_i(x) k_i(x, x') g_i(x') - \sum_{i,k} g_i(x) k_{i,T}(x)^T D_i K_T^{-1} D_k k_{k,T}(x')^T g_k(x')$
- $\boldsymbol{var_{entire}(x)} = k_T(x, x)$
- $= \sum_i g_i(x) k_i(x, x) g_i(x) - \sum_{i,k} g_i(x) k_{i,T}(x)^T D_i K_T^{-1} D_k k_{k,T}(x)^T g_k(x)$
- $= \sum_i g_i(x) k_i(x, x) g_i(x) - \sum_{i,k} z_i^T K_T^{-1} z_k$
- $\boldsymbol{= \sum_i g_i(x) k_i(x, x) g_i(x) - \sum_{i,k} z_i^T \left(\sum_j D_j K_{j,T} D_j\right)^{-1} z_k}$
- Where for a fixed x, $z_i = D_k k_{k,T}(x)^T g_k(x)$
- $K_T = \left[k(x^j, x^k)\right]_{j,k} = \sum_i \left[g_i(x^j)\, k_i(x^j, x^k) g_i(x^k)\right]_{j,k} = \sum_i D_i K_{i,T} D_i$
- $D_i = diag([g_i(x^1), g_i(x^2), \dots, g_i(x^m)])$

Long long long story…

- $k_{i,T}(x, x') = k_i(x, x') - k_{i,T}(x)^T K_{i,T}^{-1} k_{i,T}(x')$
- $var_i(x) = k_{i,T}(x, x)$
- $= k_i(x, x) - k_{i,T}(x)^T K_{i,T}^{-1} k_{i,T}(x)$
- $\boldsymbol{var(x)} = \sum_i g_i(x) var_i(x) g_i(x)$
- $= \sum_i g_i(x) k_i(x, x) g_i(x) - \sum_i g_i(x) k_{i,T}(x)^T K_{i,T}^{-1} k_{i,T}(x) g_i(x)$
- $= \sum_i g_i(x) k_i(x, x) g_i(x) - \sum_i \boldsymbol{g_i(x) k_{i,T}(x)^T D_i} D_i^{-1} K_{i,T}^{-1} D_i^{-1} \boldsymbol{D_i k_{i,T}(x) g_i(x)}$
- $= \sum_i g_i(x) k_i(x, x) g_i(x) - \sum_i \boldsymbol{z_i^T D_i^{-1} K_{i,T}^{-1} D_i^{-1} z_i}$
- $\boldsymbol{= \sum_i g_i(x) k_i(x, x) g_i(x) - \sum_i z_i^T B_i^{-1} z_i}$

Conti.

- $\boldsymbol{var_{entire}(x)} = \sum_i g_i(x) k_i(x, x) g_i(x) - \sum_{i,k} z_i^T \left(\sum_j D_j K_{j,T} D_j\right)^{-1} z_k$
- $= \sum_i g_i(x) k_i(x, x) g_i(x) - \sum_{i,k} z_i^T \left(\sum_j B_j\right)^{-1} z_k$
- $\boldsymbol{var(x)} = \sum_i g_i(x) k_i(x, x) g_i(x) - \sum_i z_i^T B_i^{-1} z_i$

- $\boldsymbol{var_{entire}(x)} - \boldsymbol{var(x)} = \sum_i z_i^T B_i^{-1} z_i - \sum_{i,k} z_i^T \left(\sum_j B_j\right)^{-1} z_k$
- Where $z_i \in R^m, B_i \in S_+^m$ a positive semi definite m by m matrix.

Conti.

- We want to prove:
  - $\sum_i z_i^T B_i^{-1} z_i - \sum_{i,k} z_i^T \left(\sum_j B_j\right)^{-1} z_k \geq 0$
  - $\Leftrightarrow \frac{\sum_i z_i^T B_i^{-1} z_i}{T} - \left(\frac{\sum_i z_i}{T}\right)^T \left(\frac{\sum_j B_j}{T}\right)^{-1} \left(\frac{\sum_i z_i}{T}\right) \geq 0$
  - $\Leftrightarrow \frac{\sum_i h(z_i, B_i)}{T} - \mathrm{h}\left(\frac{\sum_i z_i}{T}, \frac{\sum_i B_i}{T}\right) \geq 0$
  - Where $h(x, Y) = x^T Y^{-1} x$ is called a matrix fractional function, and is **convex** on $dom\ h = \boldsymbol{R}^m \times S_+^m$

Conti.

- Therefore, by Jenson's inequality on convex function h, the inequality holds.
- $\frac{\sum_i h(z_i, B_i)}{T} - \mathrm{h}\left(\frac{\sum_i z_i}{T}, \frac{\sum_i B_i}{T}\right) \geq 0$

Conti.

- Therefore, by Jenson's inequality on convex function h, the inequality holds.
- $\frac{\sum_i h(z_i, B_i)}{T} - \mathrm{h}\left(\frac{\sum_i z_i}{T}, \frac{\sum_i B_i}{T}\right) \geq 0$
- i.e. $\boldsymbol{var_{entire}(x)} - \boldsymbol{var(x)} \geq 0$
- i.e. $\boldsymbol{var_{entire}(x)} \geq \boldsymbol{var(x)}$

Conclusion

- It implies running individual Gaussian process regression helps on reducing the uncertainty.

Next step

- Our problem is a two-stage problem:
  - 1. Learn the linear approximation $g_i(v)$
  - 2. Approximate $f(x) = \sum_i g_i(v) f_i(v)$ and find the optimal solution

Next step

- Our problem is a two-stage problem:
  - 1. Learn the linear approximation $g_i(v)$
  - 2. Approximate $f(x) = \sum_i g_i(v) f_i(v)$ and find the optimal solution

Next step

- Our problem is a two-stage problem:
  - 1. Learn the linear approximation $g_i(v)$
  - 2. Approximate $f(x) = \sum_i g_i(v) f_i(v)$ and find the optimal solution
  - 3. Fit this method to our context…